Research Article

Chunsheng Jiang*

An orbit determination method of spacecraft based on distribution regression

https://doi.org/10.1515/astro-2021-0021
Received Nov 14, 2021; accepted Dec 02, 2021

Abstract: A new method of orbit determination (OD) is proposed: distribution regression. The paper focuses on the process of using sparse observation data to determine the orbit of the spacecraft without any prior information. The standard regression process is to learn a map from real numbers to real numbers, but the approach put forward in this paper is to map from probability distributions to real-valued responses. According to the new algorithm, the number of orbital elements can be predicted by embedding the probability distribution into the reproducing kernel Hilbert space. While making full use of the edge of big data, it also avoids the problem that the algorithm cannot converge due to improper initial values in precise OD. The simulation experiment proves the effectiveness, robustness, and rapidity of the algorithm in the presence of noise in the measurement data.

Keywords: distribution regression; orbit determination; kernel embedding; reproducing kernel Hilbert space

1 Introduction

In the past decade, with the advances of micro-satellite technology and the reduced costs of entering space, the number of space missions saw a rapid growth. Over time, the number of space debris grow steadily, and thus OD of these debris is also crucial to minimizing the risk of collision between active spacecraft and space debris. The current OD technology can obtain precise results, but there are many restrictions on the OD for small spacecraft with high initial uncertainty in position and non-cooperative targets. For cooperative targets, radar or GPS-based systems can work for OD. However, the radar requires low initial uncertainty of the orbit, and GPS is only suitable for low earth orbit satellites which must be equipped with a corresponding GPS system. Geostationary satellites known as GEOs play a significant role in the regional satellite navigation system; Nevertheless, due to absence of GPS systems, the OD of GEOs must rely on the measurement of ground stations. Meanwhile, due to limited observation conditions, it is likely that only a very small amount of data is available for the OD of small objects like debris. Therefore, a study to address these challenges is urgently needed.

One common approach for OD is based on the least square method, which fits by minimizing the residuals of observations and predictions. However, this method requires good initial estimates. When the observation is noisy or the system is highly nonlinear, the method will not converge (Vallado 2001; Milani and Gronchi 2010; Lee 2005). In such case, the orbit cannot be determined. In addition, the initial OD methods such as Gauss method and Laplace method only use a small number of observation points to determine the orbit and cannot make full use of the advantages of big data. As a result, the precision is low.

In recent years, machine learning has become more widely applied and its application scenarios are increasing (Izzo *et al.* 2019; Jiang *et al.* 2020). Machine learning usually targets two issues, namely classification problems and regression problems. A traditional solution to regression problem is learning a mapping of real numbers to real numbers. However, in real applications, complex problems and data often have multiple input data patterns such as probability distribution and function. In such case, the traditional regression models cannot tackle the obstacle. Some research has been done on different probability distribution problems. Wang *et al.* (2009) studied the problem of computing the similarity of bags of samples and designed a parametric model to estimate the similarity based on the obtained parameters. However, this method is limited by its over-simplified model assumed. Therefore, it is difficult to have a feasible model or very likely to face a huge amount of calculation in real scenarios. Also, a couple of nonparametric approaches were put forward (Póczos *et al.* 2012; Györfi *et al.* 2002; Wen *et al.* 2020). Oliva *et al.* (2014a,b) estimated the distribution density as an intermediate process,

Corresponding Author: Chunsheng Jiang: State Key Laboratory of Astronautic Dynamics, China Xi'an Satellite Control Center, Xi'an 710043, China; Email: csjiang1990@163.com

and then analyzed the distribution density. However, under this framework, the convergence speed was very slow for high-dimensional space scenarios (Wasserman 2006).

Distribution regression, a new machine learning method, resolves the regression problem from probability distribution to real value by embedding the distribution into the reproducing kernel Hilbert space. The concept of distribution regression was first proposed by Póczos *et al.* (2013) and a detailed mathematical analysis was made by Szabó *et al.* (2016). Several studies were carried out on distributions. Ferraty and Vieu (2006) proposed a study that the input data was a function different from the traditional finite-dimensional features which was called function regression. There are two main differences between the distribution regression and the function regression. First, the input of the distribution regression P is a probability measure in the $\mathbb{R}^k$ space rather than a one-dimensional function. Second, more importantly, the covariate P cannot be observed directly. Instead, a sample from P is observed, which means that we have a regression model with measurement error. Some studies have shown that the use of kernel embedding or its various extended forms of kernel methods to solve the distribution regression problem is an effective way (Yoshikawa *et al.* 2014). Distribution regression was used to analyze the public support for 2012 and 2016 presidential candidates during election process (Flaxman *et al.* 2015, 2016). A mixed model was proposed for the multi-instance problem to estimate the optical depth of the atmosphere based on the analysis of spatial multi-atlas satellite images (Wang *et al.* 2011). Szabó *et al.* (2015) reproduced this problem and significantly improved the prediction accuracy via distribution regression method in comparison to the traditional density function method. Although it is a common approach to resolve distribution regression problems, the kernel method suffers the disadvantages of large amount of calculation and long calculation time.

This paper, without making a priori assumption about the orbit, uses distribution regression method to determine the orbit for spacecraft with the range-only information of the ground station. The proposed method has no requirement for initial conditions. Also, sparse data is used to determine the orbit which traditional algorithm cannot deal with. Meanwhile, a weighted Fastfood algorithm is proposed to improve efficiency of the kernel trick which is often used to solve the distribution regression problems.

2 Problem setup

In most cases, using ground stations for OD is essentially a process to track an object with a network of sensors. For spacecraft i, its orbit elements Γ are drawn from the probability distribution P_Γ. Although the form of the probability distribution cannot be directly obtained, the observation data X that conforms to the probability distribution is available. In the scenario of OD with range-only data, we use P_Γ to represent the spacecraft's prior distribution. F is the actual position, velocity, and other orbital information of the spacecraft in a noise-free environment during the operation of the spacecraft, and X is the observation obtained from the ground station containing measurement noise. We hope that the proposed algorithm can determine the orbital elements of the test orbit by the observation information $\{X_i\}_{n=1}^{n_T}$ of n_T spacecraft. It is expected that the orbital elements obtained by the algorithm are closest to the actual ones. The spacecraft is not observed at a certain point only, but over a continuous period. Therefore, the OD of the spacecraft is regarded as a distribution regression problem, which takes the orbital measurement data as input, the orbital elements as output, and ignores the dynamic model. Then, a map from measurement data to orbit elements is learned. In OD, the mapping has strong nonlinearity, and the length of each observation is different. Therefore, the kernel mean embedding method is a feasible solution.

This paper is inspired by the study of Sharma and Cutler (2018). The main differences between this paper and the literature (Sharma and Cutler 2018) are as follows: First, the OD is performed under two stations and three stations with sparse range data. Second, this paper improves the traditional kernel method with an overloaded calculation storage and long calculation time. By embedding each distribution into a finite-dimensional representative vector, the calculation of the most time-consuming $\mathbf{K}$ matrix in the traditional kernel method is replaced. For details, see Section 3.3. Last, this paper compares the results of different random noise and system noise in the experiment to verify the robustness of the algorithm.

3 Distribution regression

3.1 Definition and objective function

In the standard regression problem, it is usually based on a given input vector of d-dimensional feature $X \in \mathbb{R}^d$ to predict its real-number label Y. The regression problem we consider in this paper is to give a set of variables

$(P_1, Y_1), \ldots, (P_m, Y_m)$, where $Y_i \in \mathbb{R}$, P_i represents the probability distribution on the compact set. Assuming the existence function f that satisfies:

$$Y_i = f(P_i) + \delta_i \tag{1}$$

where δ_i represents a noise variable with a mean value of 0. We hope to construct the mapping relationship between the probability distribution and the real numbers according to the given probability distribution. The difficulty is that we do not observe P_i directly, but rather we can only get the sampling observation points. For a series of sequences with N_i sampling observation points, it satisfies independently and identically distributed (iid.)

$$x_{i,1}, \ldots, x_{i,N_i} \sim P_i \tag{2}$$

Therefore, the obtained sample data are $\{(x_1, y_1), \ldots, (x_l, y_l)\}$, where $x_i = \{x_{i,1}, \ldots, x_{i,N_i}\}$ represents the i-th observation sample sequence, and y_i represents the orbital elements corresponding to the sample sequence. Our goal is to predict a new label y_{l+1} based on a new batch of sampling points x_{l+1} drawn from a new probability distribution P_{l+1}. The model representation is shown in Figure 1 (Oliva *et al.* 2014a):

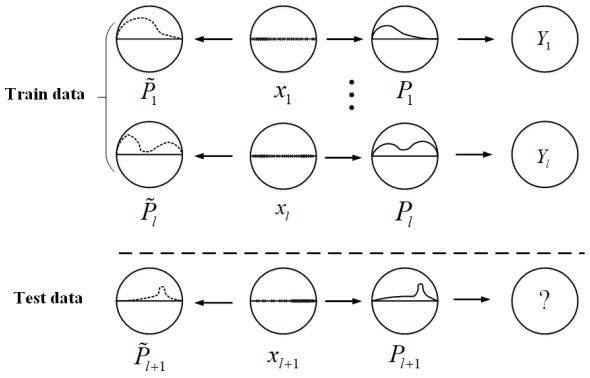


Figure 1. A graphical representation of distribution regression: the data we get is $\left\{\left(\{x_{i,n}\}_{n=1}^{N_i}\right), y_i\right\}_{i=1}^{l}$, where the label y_i only depends on the probability distribution P_i. As $\{x_{i,n}\}_{n=1}^{N_i}$ contains noise, we can only get the empirical distribution $\tilde{P}_i$ rather than P_i based on the input

3.2 Reproducing kernel Hilbert space and kernel mean embedding

In this section, we will discuss how to solve the distribution regression problem through a kernel mean embedding method. We use the kernel method to solve the problem of mapping two probability distributions to the reproducing kernel Hilbert space in which the kernel methods can be extended to probability measures (Gretton *et al.* 2012). The kernel mean embedding owes its success to a positive definite function commonly known as the kernel function. The kernel function has become popular in the machine learning community for more than 20 years.

Definition 1: Kernel function. Let $\mathcal{X}$ denote a non-empty set. The function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel function if and only if there is a Hilbert space $\mathcal{H}$ and a feature map $\psi : \mathcal{X} \to \mathcal{H}$, so that for $\forall x, x' \in \mathcal{X}$:

$$k(x, x') := \langle \psi(x), \psi(x') \rangle_{\mathcal{H}} \tag{3}$$

The kernel functions can be applied to any learning algorithm as long as they can be expressed entirely in terms of a dot product $\langle x, y \rangle$. The inner product is generally used to indicate the similarity of two points. In low-dimensional space, the calculation of the inner product is relatively easy; however, in most cases, the low-dimensional space cannot effectively distinguish the sample points. Eq. (4) expresses a mapping from two-dimensional space to three-dimensional space:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \to \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix} = \psi(x) \tag{4}$$

However, in order to solve the problem, we often need to map to a higher dimensional or infinite dimensional space. The nonlinear mapping ψ will become more complicated, and the amount of calculation will increase sharply. Fortunately, in many cases, we do not need to represent ψ explicitly as long as kernel methods have access to k. Eq. (3) shows that the calculation of the inner product in the high-dimensional space can be replaced by the kernel operation of the original space.

The idea of kernel mean embedding is to extend the feature map ψ to the space of probability distributions by representing each distribution $\mathbb{P}_x$ as a mean function:

$$\mu[\mathbb{P}_x] := \mathbb{E}_x[k(x, \cdot)] \tag{5}$$

Specifically, the mean embedding is an element defined on the Hilbert space, where k is the kernel function. When the selected kernel function is the characteristic kernel, the mean embedding of Eq. (5) retains all the information of the distribution.

Definition 2: Characteristic kernel. For the Eq. (5), when the feature map $\mu : \mathbb{P} \to \mu_{\mathbb{P}}$ of the probability distribution is injective, the kernel function k can be called a characteristic kernel. Only when the selected kernel is a characteristic

kernel, can it be concluded $||\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}|| = 0$ if and only if $\mathbb{P} = \mathbb{Q}$.

According to Definition 2, the kernel mean representation captures all information about the distribution (Fukumizu *et al.* 2004). In most cases, we cannot get the accurate form of the probability distribution , but only the sample points with noise. For a dataset containing n sampling points, we cannot get the accurate mean mapping $\mu_{\mathbb{P}}$ but only its empirical estimate:

$$\hat{\mu}_{\mathbb{P}} := \frac{1}{n}\sum_{i=1}^{n} k(x, \cdot) \tag{6}$$

Song (2008) proved that for $\forall f \in \mathcal{H}$, if $||f||_{\infty} \leq 1$ was satisfied, the mean mapping of the empirical distribution converged to the mean mapping of the true distribution. This ensures that the method of predicting the label through the learning of the empirical probability distribution is effective and feasible.

Let X denote the set of Borel probability measures on $\mathcal{X}$. Then the set of mean embedding X_μ can be expressed as:

$$X_\mu = \mu(X) = \{\mu_x : x \in X\} \subseteq \mathcal{H} \tag{7}$$

We complete the calculation of the distribution regression through a two-stage mapping. That means to map the distribution data $x \in X$ to X_μ first, and then map the mean embedding to the real-number label Y by defining the reproducing kernel Hilbert space function f. The process can be expressed as

$$x \in X \xrightarrow{\psi} X_\mu (\subseteq \mathcal{H}) \xrightarrow{f \in \mathcal{H}} \mathbb{R}$$

In order to prevent over-fitting caused by the complicated model, a regularization term is added, and the model is established through ridge regression. The objective function is defined as:

$$f_x^{\lambda} = \arg\min_{f \in \mathcal{H}} \frac{1}{l}\sum_{i=1}^{l} [f(\mu_{x_i}) - y_i]^2 + \lambda \|f\|_{\mathcal{H}}^2 \tag{8}$$

where λ is the regularization coefficient. Since we do not get all samples x, but only a part of samples $\hat{x}$, the optimization function is:

$$f_{\hat{x}}^{\lambda} = \arg\min_{f \in \mathcal{H}} \frac{1}{l}\sum_{i=1}^{l} \left[f\left(\mu_{\hat{x}_i}\right) - y_i\right]^2 + \lambda \|f\|_{\mathcal{H}}^2 \tag{9}$$

Normally, it is difficult to obtain an accurate form of ψ in Eq. (7), resulting in that μ_{x_i} cannot be expressed exactly. Nevertheless, as explained above, calculating the inner product in a high-dimensional space is equivalent to calculating the kernel function in the original space. As a result, it can be obtained from Eq. (9) that given train samples $\{x_{i,n}\}_{i=1}^{N}$ and their labels $\{y_1, \ldots, y_l\}$, the prediction for a new test distribution l+1 is (Poggio and Shelton 2002):

Table 1. Different kernel function forms

kernel function	expression
$k_G(a, b)$	$e^{-\frac{\|a-b\|_2^2}{2\sigma^2}}$
$k_e(a, b)$	$e^{-\frac{\|a-b\|_2}{2\sigma^2}}$
$k_C(a, b)$	$\frac{1}{1+\frac{\|a-b\|_2^2}{\sigma^2}}$
$k_p(a, b)$	$(\langle a, b\rangle + \sigma)^p$
$k_r(a, b)$	$1 - \frac{\|a-b\|_2^2}{\|a-b\|_2^2+\sigma}$

$$\left(f_{\hat{x}}^{\lambda} \circ \mu\right)(l+1) = [y_1, \ldots, y_l]\left(\mathbf{K} + l\lambda \mathbf{I}_l\right)\mathbf{k} \in \mathbb{R} \tag{10}$$

$$\mathbf{K} = \left[K\left(\mu_{\hat{x}_i}, \mu_{\hat{x}_j}\right)\right] \in \mathbb{R}^{l \times l},$$

$$\mathbf{k} = \left[K\left(\mu_{\hat{x}_1}, \mu_{l+1}\right); \ldots; K\left(\mu_{\hat{x}_l}, \mu_{l+1}\right)\right] \in \mathbb{R}^{l}$$

where $\circ$ represents the combination of two operations. In Eq. (10), the $\mathbf{K}$ matrix which is called Gram matrix involves the calculation of two kernel functions. The calculation of $K\left(\mu_{\hat{x}_i}, \mu_{\hat{x}_j}\right)$ lies in the selection of the kernel function K. For a variety of forms of K, there is no significant difference in accuracy. In order to simplify the calculation, the linear kernel of K is selected in the experiment, namely $K\left(\mu_{\hat{x}_i}, \mu_{\hat{x}_j}\right) = \left\langle \mu_{\hat{x}_i}, \mu_{\hat{x}_j} \right\rangle$. Eq. (6) can be further written as:

$$\begin{aligned} K\left(\mu_{\hat{x}_i}, \mu_{\hat{x}_j}\right) &= \left\langle \mu_{\hat{x}_i}, \mu_{\hat{x}_j} \right\rangle \\ &= \frac{1}{N_i}\frac{1}{N_j}\sum_{i=1}^{N_i}\sum_{j=1}^{N_j} \left\langle \psi(x_i), \psi\left(x_j\right)\right\rangle \end{aligned} \tag{11}$$

From Eq. (3), we get $\left\langle \psi(x_i), \psi\left(x_j\right)\right\rangle = k\left(x_i, x_j\right)$, where k is the second kernel function we need to select. There are many choices for k here, including Gaussian kernel, exponential kernel, Cauchy kernel, polynomial kernel and rational quadratic kernel. The expressions of different kernel functions are listed in Table 1.

3.3 Fast estimation of kernel function

Despite the kernel method's success, what makes it difficult to use in many large-scale problems is due to the fact that computing the Gram matrix is typically expensive, especially at prediction time. In Eq. (10), if the train data contains N distributions and each distribution has n sampling points, the calculation time complexity of $\mathbf{K}$ is $O(N^2n^2)$. As the dataset grows, the expense for calculating $\mathbf{K}$ also grows. Generally, both N and n are relatively large, and calculating $\mathbf{K}$ is very time-consuming. If the feature map ψ can be estimated, the value of mean embedding

μ_{x_i} can be directly calculated, which will greatly reduce the amount of calculation. Speeding up kernel methods has been a research focus for many years. A method of introducing a sparse matrix to calculate the **K** matrix was introduced, but fundamentally it did not avoid the kernel operation between every two samples (Feng *et al.* 2019). The random Fourier method (Rahimi and Recht 2008), which aimed to find a low-dimension function to map the original D-dimensional data to the d-dimensional random feature space. The inner product of the two features in the random feature space is approximately equal to kernel function. This method decreased the computational time complexity to $O(nd)$, which means that it was irrelevant to the number of distributions N. Le *et al.* (2013) combined the Gaussian diagonal Walsh-Hadamard matrix and the dense Gaussian matrix and proposed a Fastfood algorithm to map the D-dimensional distribution to the d-dimension. The matrices that we consider are parameterized by a product of diagonal and simple matrices

$$\phi(x) = \sqrt{d}e^{i[Vx]} \qquad V = \frac{1}{\sigma\sqrt{D}}SHG\Pi HB \tag{12}$$

where S, B, G are all diagonal random matrices. S is a non-negative random scaling matrix, whose form is determined according to the kernel function to be approximated. More specifically, B has random $\{\pm 1\}$ entries on its main diagonal; G has random Gaussian entries; H is the Walsh-Hadamard matrix; Π is a random permutation matrix to ensure that the two H matrices are uncorrelated; V is then used to compute the feature map $\phi(x)$ which further shortens the calculation time to $O(n\log d)$.

The improvement process of the two-stage sampling algorithm based on the Fastfood algorithm is as follows:

Algorithm Two-step sampling algorithm based on Fastfood

Input: train data $\left\{\left(\{x_{i,n}\}_{n=1}^{N_i}\right), y_i\right\}_{i=1}^{l}$, test data $\{x_{t,j}\}_{j=1}^{N_t}$

Calculate V
for i in l:
  for j in N_i:
    Calculate $\phi(x)$
    Calculate $\mu_{\hat{x}_i} = \frac{1}{N_i}\sum_{j=1}\phi\left(x_{i,j}\right)$
Calculate $\mu_t = \frac{1}{N_t}\sum_{j=1}\phi\left(x_{t,j}\right)$
Calculate $\mathbf{K} = \left[K\left(\mu_{\hat{x}_i}, \mu_{\hat{x}_j}\right)\right]$, $\mathbf{k} = \left[K\left(\mu_{\hat{x}_1}, \mu_t\right); \ldots; K\left(\mu_{\hat{x}_l}, \mu_t\right)\right]$
Calculate $\left(f_{\hat{z}}^{\lambda} \circ \mu\right)(t)$ in Eq. (10)

Output: $\left(f_{\hat{z}}^{\lambda} \circ \mu\right)(t)$

Though this method is suitable for distribution data, it only fits one-dimensional feature samples. Our experiment involves multiple stations which means each sample contains multiple features. Therefore, we propose to perform a weighted improvement for each feature. The kernel mean of each sample after mapping is $\hat{\mu} = \frac{\sum_i w_i \psi_i}{\sum_i w_i}$, where w_i represents the weight of the i-th feature, and ψ_i represents the feature map of the i-th feature.

4 Simulation experiment and discussion

4.1 Sampled data generation

The experiment designed in this paper uses range-only data from three ground stations to determine the orbit of GEO spacecraft. For comparison, we also add an experiment with just two ground stations. At the same time, we use sparse observation data of only two sample points in one spacecraft period. The current technology cannot deal with the OD in such case. The coordinates of the three stations selected are $[46.8, 130.32, 0.101]^{\mathrm{T}}$, $[39.505, 75.929, 1.255]^{\mathrm{T}}$, $[18.313, 109.311, 0.018]^{\mathrm{T}}$ corresponding to latitude (degree), longitude (degree), elevation (km). The prior distributions of the GEO spacecraft orbit elements are:

$$\begin{aligned} a &\sim U(42160.7, 42170.7)\ \mathrm{km} \\ e &\sim U(0.0001, 0.001) \\ i &\sim U(0, 0.05)\ \mathrm{deg} \\ \Omega &\sim U(220, 230)\ \mathrm{deg} \\ \omega &\sim U(0, 10)\ \mathrm{deg} \\ M &\sim U(45, 50)\ \mathrm{deg} \end{aligned}$$

A point worth mentioning is that the prior distributions are only used to produce train data but not as a basis for prediction. Within the given range above, 2000 orbits are randomly generated. High Precision Orbit Propagator is adopted as the orbit propagator. The spacecraft's area-to-mass ratio is $0.02\ \mathrm{m}^2/\mathrm{kg}$. The Sun and the moon are considered as the third body perturbance. The propagator time is 24 hours. The stations observe each orbit and take one observation point every 1 hour. We randomly select 1200 orbits as the train set, 400 as the validation set to determine the appropriate kernel function parameters σ and the regularization coefficient of the regression function λ, and the remaining 400 are used as the test set to verify the performance of the algorithm. So, the shape of train data is (1200,24,3). Since the right ascension of ascending node Ω, the argument of perigee ω, and the mean anomaly M vary

in the range of $[x_1, 360] \cup [0, x_2]$, which is not a compact set. Therefore, the sine or cosine function is used to process them. After the transformation, the new orbital elements $(a, l, e_x, e_y, i_x, i_y)$ are defined as follows:

$$\begin{aligned} a &= a \\ l &= \Omega + \omega + M \\ e_x &= e\cos(\Omega + \omega) \\ e_y &= e\sin(\Omega + \omega) \\ i_x &= i\cos(\Omega) \\ i_y &= i\sin(\Omega) \end{aligned}$$

where l is called phase, which equals the sum of the right ascension of ascending node, the argument of perigee and the mean anomaly.

4.2 Result analysis

We predict the six transformed elements of the orbit respectively. In the choice of kernel function, we compare the five kernel functions in Table 1. Under the optimal parameter conditions, the exponential kernel effect is poor, while the accuracy of the rest four kernel functions shares no obvious difference. As a result, we choose the most commonly used Gaussian kernel as the kernel function k. A grid search method is adopted to search the optimal parameters whose domain is $(\lambda, \sigma) \in \left\{2^{-60}, 2^{-59}, \ldots, 2^{0}\right\} \times \{100, 200, \ldots, 5000\}$. The final optimal parameters corresponding to each orbital elements are shown in Table 2:

Table 2. Optimal parameters

orbital elements	σ	λ
a	2600	2.910e−13
l	300	3.638e−12
e_x	100	1.776e−15
e_y	100	1.776e−15
i_x	100	1.776e−15
i_y	100	1.776e−15

Table 3. Prediction accuracy of semi-major axis

MSE in semi-major axis	two stations	three stations
no noise	52.203m	42.862m
10m random noise	52.241m	43.926m
100m random noise	54.953m	47.491m
5m system noise	55.542m	40.735m
5m system noise +10m random noise	55.104m	46.276m
5m system noise +100m random noise	55.817m	44.788m

For comparison, in the weighted Fastfood algorithm, we also use the same parameters as the Gaussian kernel σ. In the experiment, the similarity of the $\mathbf{K}$ matrix calculated by the two algorithms is above 98% through comparison, indicating that the accuracy of the kernel function prediction is guaranteed by the improved algorithm. For the prediction of the orbital elements, experiments are conducted to verify the accuracy of OD under different noise levels. Table 3 to Table 6 show the error levels of the six transformed orbital elements under different noise levels. As e_y and i_y are similar to e_x and i_x in error and principle, they are not reflected in the table to save space. Each experiment randomly divides the train set and the test set, and 10 experiments are conducted for an average result:

It can be seen from the above table that adding system noise does not change the probability distribution, so this method can also well overcome the impact of measure-

Table 4. Prediction accuracy of phase

MSE in phase	two stations	three stations
no noise	0.00670°	0.00201°
10m random noise	0.00665°	0.00233°
100m random noise	0.00669°	0.00292°
5m system noise	0.00667°	0.00217°
5m system noise +10m random noise	0.00672°	0.00215°
5m system noise +100m random noise	0.00675°	0.00297°

Table 5. Prediction accuracy of e_x

MSE in e_x	two stations	three stations
no noise	5.231e−5	5.859e−6
10m random noise	5.224e−5	1.034e−5
100m random noise	5.248e−5	1.278e−5
5m system noise	3.526e−5	5.496e−6
5m system noise +10m random noise	4.164e−5	1.274e−5
5m system noise +100m random noise	5.217 e−5	4.710e−5

Table 6. Prediction accuracy of i_x

MSE in i_x	two stations	three stations
no noise	0.0105°	3.064e−4°
10m random noise	0.0106°	0.00220°
100m random noise	0.0110°	0.00233°
5m system noise	0.0106°	5.120e−4°
5m system noise +10m random noise	0.0107°	0.00223°
5m system noise +100m random noise	0.0107°	0.00234°

ment errors caused by system noise. In subsequent experiments, the magnitude of the noise is gradually increased. It is found that when the noise reaches the level of 500m, the prediction accuracy will significantly decrease, and in real applications the measurement error generally does not reach this magnitude. This also verifies the robustness of our method to noise.

In order to get more convincing results, we duplicate the experiment for 10 cycles. We compare the results of data from three stations and data from two stations under different levels of noise. The only difference between the two types of data is that the data from the third station are removed. The results of 4000 test samples are shown in Figures 2–5 which the comparison of the predicted values of the orbital elements after each correction with the real values under the condition of 5m system noise and 10m Gaussian noise.

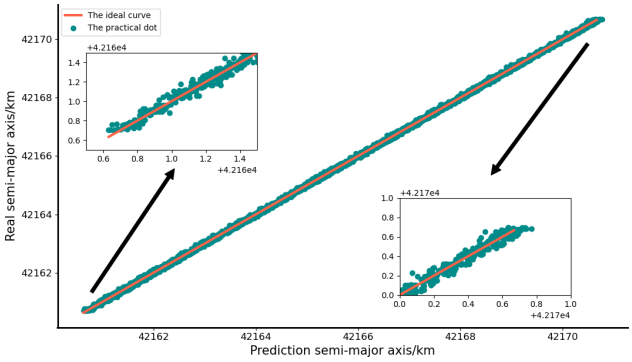


Figure 2. The predicted and real values of semi-major axes of 400 test samples for 10 cycles: The predicted values of the semi-major axes come from range-only data of 3 ground stations. The predicted values almost fit the ideal cure with a slope of 1. The lower limit and the upper limit are specially displayed to express the error of predicted and real values

Figure 3 and Figure 4 express the accuracy differences of three and two ground stations. The result shows that a satisfactory accuracy is achieved by our method in the case of three ground stations. The outcome is also quite good even for the scenario of two stations.

An experiment is conducted when only two samples are collected in a spacecraft's period. In other words, we get a sample every 12 hours in a day. This means we can only determine the spacecraft's orbit with two range-only samples. The shape of train data turns into (1200,2,3) and (1200,2,2). We test the OD ability of the proposed method with sparse data. Traditional OD methods cannot deal with this situation. As we have trained a model, we can predict the orbital elements. The results are shown in Figure 6.

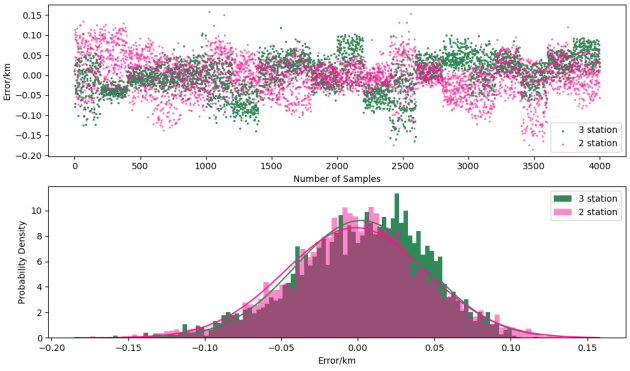


Figure 3. The errors of predicted and real values of semi-major axes and the probability density of the errors: The upper figure shows that the errors of data from three stations are closer to zero compared with the case of two stations. Meanwhile, the lower figure shows the difference of probability density curves. Both distributions of errors are Gaussian. It can be seen that the green curve has a higher peak at 0 error which means the predicted values of semi-major axes of three stations are more accurate

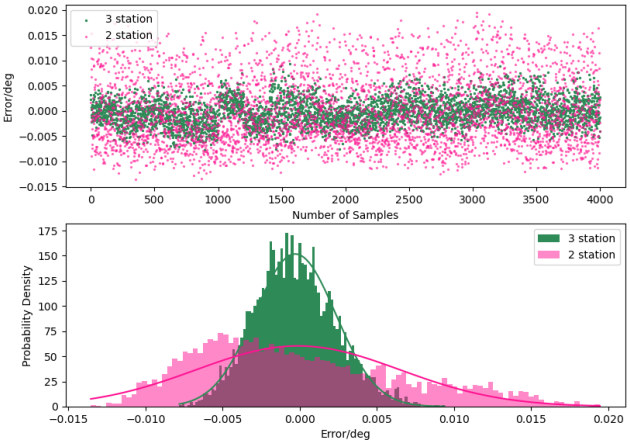
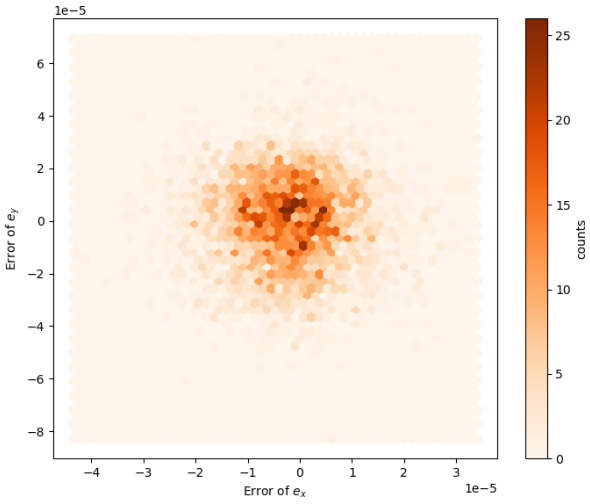


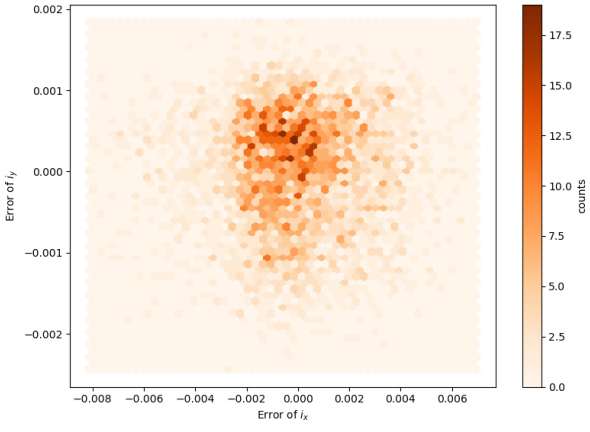
Figure 4. The errors of predicted and real values of phases and the probability density of the errors: The upper figure shows that the phase errors of three stations are in most cases less than 0.005 deg while the errors of two stations reach 0.02 deg. The lower figure clearly illustrates the probability densities of the two datasets. Both distributions of errors are Gaussian. The green curve's peak is at the error of 0; by contrast, the orange curve is flatter, and its peak is much lower than that of the green. It means the predicted phases of two stations are of low accuracy

Though the OD is finished by only two samples, a conclusion is obtained that most of the predictions have a bias in semi-major axes. A possible reason we can infer is that the samples contain system noise and random noise. To verify our guess, we remove the system noise added in the previous experiments. We draw the difference again in Figure 7.

Figure 7 illustrates that the mean value of the probability density moves to zero after removing the system noise



(a)



(b)

Figure 5. a) The errors of e_x and e_y from three stations: The same level of errors is expressed in e_x and e_y. b) The errors of i_x and i_y from three stations: The error of transformed inclination is asymmetry. The figure shows i_y has a little negative bias

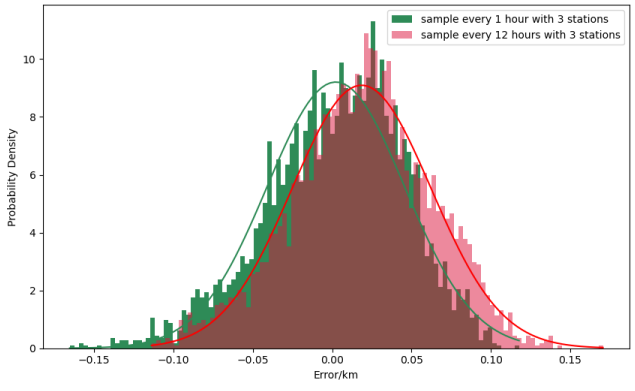


Figure 6. The probability density of the errors under three ground stations: The green curve has a peak in error 0, while the red curve has a bias in error peak

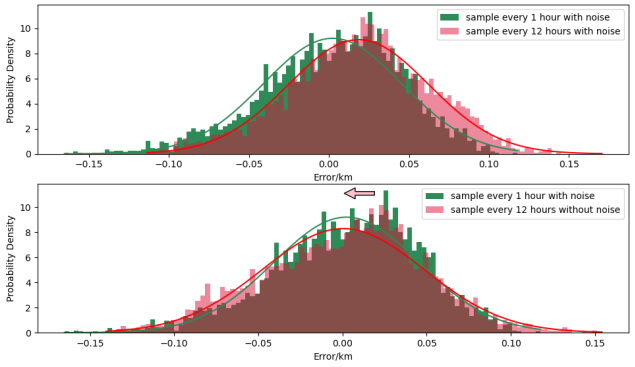


Figure 7. The probability density of the errors before and after removing noise: After removing the system noise of the sparse data experiments, the peak move to error 0

when using the sparse data. Therefore, the method can also be effective with sparse data if the peak can be moved to zero. Fortunately, for machine learning, it is an easy case for the bias to be leant if the relationship between the level of system noise and the bias is given. This is not discussed in our paper.

At last, a time comparison is also conducted in simulations. The computer used in the experiment is configured with i9-9900CPU@3.10GHz 16G memory. For the train set containing 1600 samples, the train time for the kernel method and the algorithm we proposed are 27.37s and 4.14s respectively. It can be seen that the speed increase of the kernel method is quite obvious: the larger the data volume, the more obvious the advantage in the calculation speed.

5 Conclusion

Based on the traditional regression problem in machine learning, this paper proposes a method for orbit determination by distribution regression. This method does not rely on the prior information of the orbit and the dynamic model of the spacecraft, but only uses the observation data. The distribution data is mapped into the reproducing kernel Hillbert space and the orbital elements are predicted directly. Meanwhile, a new weighted Fastfood method is proposed to improve the computational efficiency of the kernel method in the distribution regression problem. Simulation experiments are carried out to verify the effectiveness to sparse data and robustness to noise of the proposed method. The results show that the predicted errors of semi-major axes of GEOs using range-only data from three ground stations and two stations are about 40m and 55m respectively under the noisy condition. The errors of phases are respectively about 0.006° and 0.002°. The semi-major axis and

phase are the most important orbital elements for the location of GEOs. The accuracy and speed are much higher and faster than the results in Sharma and Cutler's study in 2018. Though the accuracy is still lower than that of the precise OD method, our method does not rely on the initial value. The approach proposed can still get a satisfactory result even with so sparce data that cannot be handled by traditional OD methods.

In the future research, we will expand our method applications to numerous debris OD with different kinds of measurement data, and work for an improved accuracy.

Funding Information: This project is funded by Equipment Pre-research Key Laboratory Fund (No.6142210200102)

Conflict of interest: The authors state no conflict of interest.

References

Feng F, Zhang Y, Li H, Fang Y, Huang Q, Tao X. 2019. A novel space-based orbit determination method based on distribution regression and its sparse solution. IEEE Access. 7:133203-133217.

Ferraty F, Vieu P. 2006. Nonparametric functional data analysis: theory and practice. New York: Springer.

Flaxman SR, Wang YX, Smola AJ. 2015. Who supported Obama in 2012? Ecological inference through distribution regression. Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2015 Aug 10-13; Sydney, Australia. Association for Computing Machinery: 289-298.

Flaxman S, Sutherland DJ, Wang YX, The YW. 2016. Understanding the 2016 US Presidential Election using ecological inference and distribution regression with census microdata. ArXiv preprint. Available from: arXiv:1611.03787.

Fukumizu K, Bach FR, Jordan MI 2004. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. J Mach Learn Res. 5(Jan):73-99.

Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola, A. 2012. A kernel two-sample test. J Mach Learn Res. 13(1):723-773.

Györfi L, Kohler M, Krzyżak A, Walk H. 2002. A distribution-free theory of nonparametric regression. Vol. 1. New York: Springer.

Izzo D, Mrtens M, Pan B. 2019. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. Astrodynamics. 3(4):287-299.

Jiang J, Zeng X, Guzzetti D, You Y. 2020. Path planning for asteroid hopping rovers with pre-trained deep reinforcement learning architectures. Acta Astronaut. 171:265-279.

Le Q, Sarlós T, Smola A. 2013. Fastfood-approximating kernel expansions in loglinear time. Proceedings of the 30th International Conference on Machine Learning; 2013 June 16-23; Atlanta (GA), USA. JMLR: 28(3):244-252.

Lee DJ. 2005. Nonlinear Bayesian filtering with applications to estimation and navigation [dissertation]. College Station: Texas A&M University.

Milani A, Gronchi G. 2010. Theory of orbit determination. Cambridge University Press.

Oliva J, Neiswanger W, Póczos B, Schneider J, Xing E. 2014. Fast distribution to real regression. Proceedings of 17th International Conference on Artificial Intelligence and Statistics; 2014 April 22-25; Reykjavic, Iceland. JMLR: 33:706-714.

Oliva J, Neiswanger W, Póczos B, Schneider J, Xing E. 2014b. Fast distribution to real regression. Artificial Intelligence and Statistics. PMLR. p.706-714.

Póczos B, Xiong L, Schneider J. 2012. Nonparametric divergence estimation with applications to machine learning on distributions. ArXiv preprint. Available from: arXiv:1202.3758.

Póczos B, Singh A, Rinaldo A, Wasserman L. 2013. Distribution-free distribution regression. Proceedings of 16th International Conference on Artificial Intelligence and Statistics. 2013 April 29-1; Scottsdale (AZ), USA. JMLR: 31:507-515.

Poggio T, Shelton CR. 2002. On the mathematical foundations of learning. Bull New Ser Am Math Soc. 39(1):1-49.

Rahimi A, Recht B. 2008. Weighted sums of random kitchen sinks: replacing minimization with randomization in learning. NeurIPS Proc. 1313-1320.

Sharma S, Cutler JW. 2018. Kernel embedding approaches to orbit determination of spacecraft clusters. ArXiv preprint. Available from: arXiv:1803.00650.

Song L. 2008. Learning via Hilbert space embedding of distributions [dissertation]. Sydney: School of Information Technologies. 127-141.

Szabó Z, Gretton A, Póczos B, Sriperumbudur B. 2015. Two-stage sampled learning theory on distributions. Proceedings of the 18th International Conference on Artificial Intelligence and Statistics; 2015 May 9-12; San Diego (CA), USA. JMLR: 38:948-957.

Szabó Z, Sriperumbudur BK, Póczos B, Gretton A. 2016. Learning theory for distribution regression. J Mach Learn Res. 17(1):5272-5311.

Wang F, Syeda-Mahmood T, Vemuri BC, Beymer D, Rangarajan A. 2009. Closed-form Jensen-Renyi divergence for mixture of Gaussians and applications to group-wise shape registration. International Conference on Medical Image Computing and Computer-Assisted Intervention. Berlin, Heidelberg: Springer. 648-655.

Wang Z, Lan L, Vucetic S. 2011. Mixture model for multiple instance regression and applications in remote sensing. IEEE Trans Geosci Remote Sens. 50(6):2226-2237.

Wen T, Zeng X, Circi C, Gao Y. 2020. Hop reachable domain on irregularly shaped asteroids. J Guid Control Dyn. 43(7):1269-1283.

Wasserman L. 2006. All of nonparametric statistics. New York: Springer.

Yoshikawa Y, Iwata T, Sawada H. 2014. Latent support measure machines for bag-of-words data classification. NeurIPS Proc. 27:1961-1969.

Vallado DA. 2001. Fundamentals of astrodynamics and applications. Vol. 12. El Segundo: Microcosm Press; Dordrecht, Boston, London: Kluwer Academic Publishers.